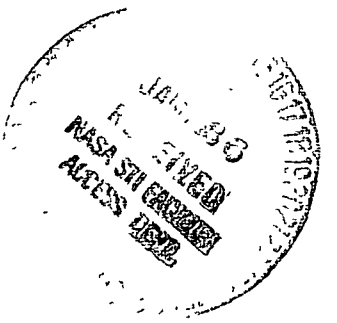
DAA/Goddard

(NASA-CR-177823) GATEWAY DESIGN SPECIFICATION FOR FIBER OPTIC LOCAL AREA NETWORKS (Computer Technology Associates, Inc.) 61 p HC A04/MF A01 CSCL 09B

N86-16991

Unclas
G3/62 15966



COMPUTER TECHNOLOGY ASSOCIATES, INC.
Denver • Washington, D C • Colorado Springs • Albuquerque • San Jose

GATEWAY DESIGN SPECIFICATION

FOR

FIBER OPTIC LOCAL AREA NETWORKS

CONTRACT NO. NAS5-28583

(Gateway Design Task)

PREPARED FOR:

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

GODDARD SPACE FLIGHT CENTER

CODE 700

GREENBELT, MARYLAND 20771

PREPARED BY:

COMPUTER TECHNOLOGY ASSOCIATES, INC.

10227 WINCOPIN CIRCLE, SUITE 208

COLUMBIA, MARYLAND 21044

AUGUST 29, 1985

TABLE OF CONTENTS

SECTION 1 - INTRODUCTION

This document is a Design Specification for a gateway to interconnect fiber optic local area networks (LAN's). This document was developed by Computer Technology Associates (CTA) Inc. for the National Aeronautics and Space Administration (NASA) Goddard Space Flight Center (GSFC) under contract number NAS5-28583.

1.1 Scope

The scope of this Design Specification is to functionally define the internetworking protocols for a gateway device that will interconnect multiple local area networks. This specification will serve as input for preparation of detailed design specifications for the hardware and software of a gateway device. General characteristics to be incorporated in the gateway such as node address mapping, packet fragmentation, and gateway routing features are described. Implementation dependent features such as dynamic versus static buffer allocation schemes are not addressed in this specification. These are to be prepared in subsequent preliminary and detailed design activities.

This Design Specification addresses a two phase approach for developing the gateway functionality. In the first phase, the gateway shall employ a connectionless oriented protocol for ISO/OSI layer 3 (i.e., network). This shall be used for an initial Demonstration capability. Then in Phase 2, the gateway layer 3 processing shall be augmented to handle Operational requirements. This shall require a connection oriented protocol

for the layer 3 in addition to the connectionless sublayer. The inclusion of a connection oriented protocol enables support for high throughput requirements such as digitized video or telemetry data. In particular, it provides for greater control by preallocating resources and using virtual circuits for data transmission.

Design specifications for the fiber optic LAN gateway address generic characteristics to be incorporated in its design. However, the gateway environment is ultimately expected to include at least three LAN's currently under development for NASA; namely:

- 100 Mbps Fiber Optic Demonstration System being developed by Sperry for NASA
- 10 Mbps fiber optic LAN being developed in-house by NASA Code 700
- token ring fiber optic LAN being developed at Johnson Space Center (JSC)

The gateway shall support internetworking between these three systems as a minimum requirement.

The remainder of this section provides a list of relevant documents for this specification. It also addresses the requirement to adhere to the principles established by the International Standards Organization (ISO) Open Systems Interconnection (OSI) seven layer reference model.

The remainder of this Design Specification consists of four additional sections which are supplemented with two appendices. In Section 2, CTA presents background information for the three LAN systems the gateway is required to support. Section 3 focuses on the environment, interfaces, and assumptions for the gateway

hardware and software. The gateway demonstration phase functionality is described in Section 4. The last section (Section 5) describes the operational internetworking functionality which shall be included in the gateway design. The rationale for the specifications presented in Sections 4 and 5 is supplied in Appendix A. In the appendix, CTA describes and compares alternative gateway architectures and recommends an architecture for implementation. Appendix B supplies the salient characteristics of the FODS HSI and RS-232 user interfaces.

1.2 Relevant Documentation

Table 1.2-1 lists those documents utilized by CTA to produce this Design Specification. Information in these documents was also supplemented by information from the Contracting Officers Technical Representative (COTR) at scheduled technical interchange and project status meetings.

1.3 Adherence to ISO/OSI Reference Model

The gateway design shall adhere to the ISO/OSI seven layer reference model. Of the seven layers (Physical, Data Link, Network, Transport, Session, Presentation, and Application), the gateway design shall incorporate layers one (Physical) through three (Network). Higher layer protocols are incorporated in the gateway only for network management purposes. Network management activities include the processing and delivery of network statistics and diagonstic reports.

TABLE 1.2-1: Relevant Documentation

Document	Source
1. Fiber Optics Demonstration System Critical Design Review June 20,21, 1984	Sperry Corporation
2. FODS System Specification March 1985	Sperry Corporation
3. Bus Interface Unit (BIU) Specification May 6, 1985	Sperry Corporation
4. FDDI Token Ring Media Access Control Draft Proposed Standard March 1, 1985	American National Standards Committee (X3T9.5 Committee)
5. High Level Protocol Study for Fiber Optic Local Area Network June 4, 1985	CTA,Inc.
6. Gateway Requirements Analysis June 7, 1985	CTA,Inc.
7. Code 700 Fiber Optic LAN Design Notes	NASA/GSFC, Code 700

SECTION 2 - BACKGROUND INFORMATION

This section provides background information on the three LAN's the gateway is required to support: the Sperry FODS, the GSFC Code 700 LAN, and the JSC LAN.

2.1 Sperry Fiber Optic Demonstration System

The Sperry Fiber Optic Demonstration System (FODS) utilizes a star configuration to interconnect a maximum of 32 nodes. It is designed to transmit data between nodes at 100 Mbps. User stations are connected to the FODS via Bus Interface Units (BIU's). A BIU is comprised of four subsystems: a High Speed Interface (HSI), a Network Interface Unit (NIU), a Front End Processor (FEP), and an RS-232 serial interface. The latter supports three independently programmable, asynchronous I/O ports. The NIU handles layer 1 and 2 (i.e., Physical and Data Link) processing. Limited functions of the Network layer (layer 3) are handled by the FEP in conjunction with the HSI and RS-232 ports to the user. The HSI is a custom high speed serial I/O port. It offers 32 Mbps synchronous serial data service for wide bandwidth user applications. Each of the three RS-232 I/O ports is a half duplex, asynchronous serial connection to the user, which support speeds up to 9600 bps.

The star topology of the FODS is such that a transmission by one BIU, via the FODS star coupler, is simultaneously received by all other BIU's on the network. With the exception of a broadcast message to all BIU's, only the addressed BIU will respond to a transmission.

Users connected to the FODS via the BIU's may be sophisticated host processors, intelligent workstations, or instruments which periodically report their measurements to another user on the network.

Packet sizes up to a maximum of 2048 bytes, including high level protocol headers, are supported by the FODS. Specifications for packet formats and sizes as they are interchanged between BIU and user are defined in the Sperry documentation referenced in Section 1.2 of this Design Specification.

2.2 NASA/GSFC Code 700 Fiber Optic LAN

The Code 700 LAN at GSFC is currently in the design stage. It is intended to be a star based LAN capable of 10Mbps transmission speeds. Up to TBD nodes will be supported by this LAN. Users are connected to the LAN via BIU's. It is intended that an HSI or RS-232 I/O port (but not both) will be available with each BIU to accommodate a user station (e.g., host computer, intelligent workstation, or instrument). Maximum packet sizes on the Code 700 LAN are smaller than those on the FODS; a maximum of 144 Bytes per packet is supported. Information on the Code 700 LAN has not been formally documented and is subject to change.

2.3 NASA/JSC Token Ring LAN

JSC is currently designing a high speed token ring LAN capable of 100Mbps transmission speeds. Data Link layer protocol for the JSC ring is defined by the Draft Proposed American National Standard "FDDI Token Ring Media Access Control" (X3T9.5

Technical Committee) March 1, 1985. One of the salient characteristics of the FDDI specification is that the maximum frame length is 4500 bytes. This is more than twice the frame size of the FODS and approximately 30 times the size of the Code 700 LAN. As such, packets originating at the JSC LAN and destined for either of the other two LAN's will very likely require that they be fragmented into smaller packets.

SECTION 3 - SCOPE

This specification defines the internetworking protocols to be implemented in the gateway. These protocols consist of two sublayers:

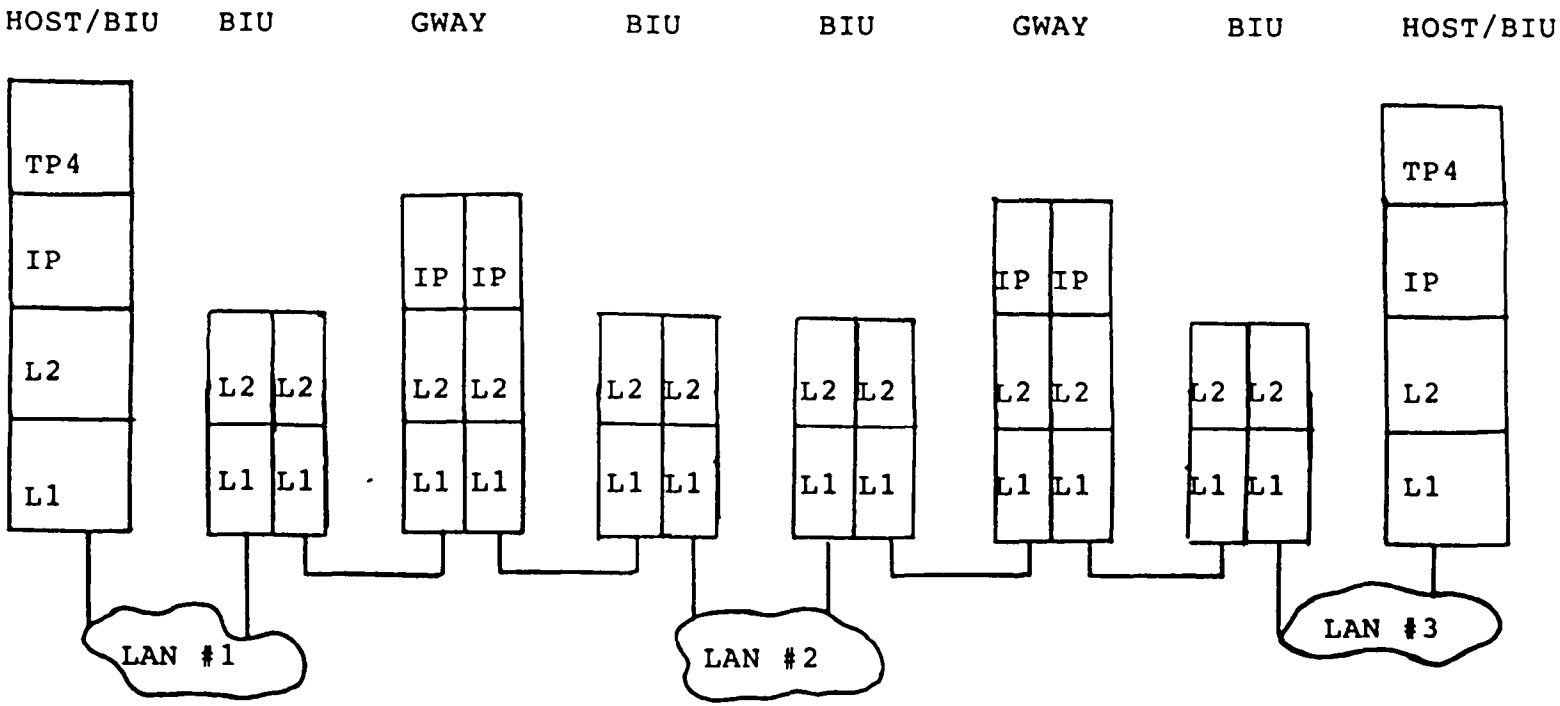
- a connection-oriented sublayer spanning homogeneous LAN's,
- a connectionless sublayer spanning heterogeneous LAN's.

As noted in Section 2, the gateway functionality shall be developed in two phases. The initial Demonstration phase shall accommodate only the connectionless sublayer. Subsequent to this, the Operational phase shall then enhance this phase with a connection oriented sublayer.

Figures 3-1a and 3-1b illustrate the gateway and its functional interrelationship with other LAN elements: BIU/host combinations, independent BIU's, and other gateways. As shown in Figure 3-1a, the Initial Demonstration Phase includes only a connectionless Internetwork Protocol sublayer. The connection oriented sublayer is null. Then, for the Operational Phase, this is augmented by a connection oriented (e.g., X.25) sublayer to facilitate virtual circuit service.

This section defines the interfaces the gateway is required to support and assumptions regarding its design. The gateway shall interface to a LAN by means of a BIU. Regardless of implementation, it shall be assumed that the BIU only handles layers 1 and 2 of the ISO/OSI 7 layer model. ISO/OSI layers 3 and above shall be handled by devices connected to the BIU (i.e., host processor, intelligent workstation, instrument, or gateway).

DEMONSTRATION PHASE PROTOCOL ARCHITECTURE

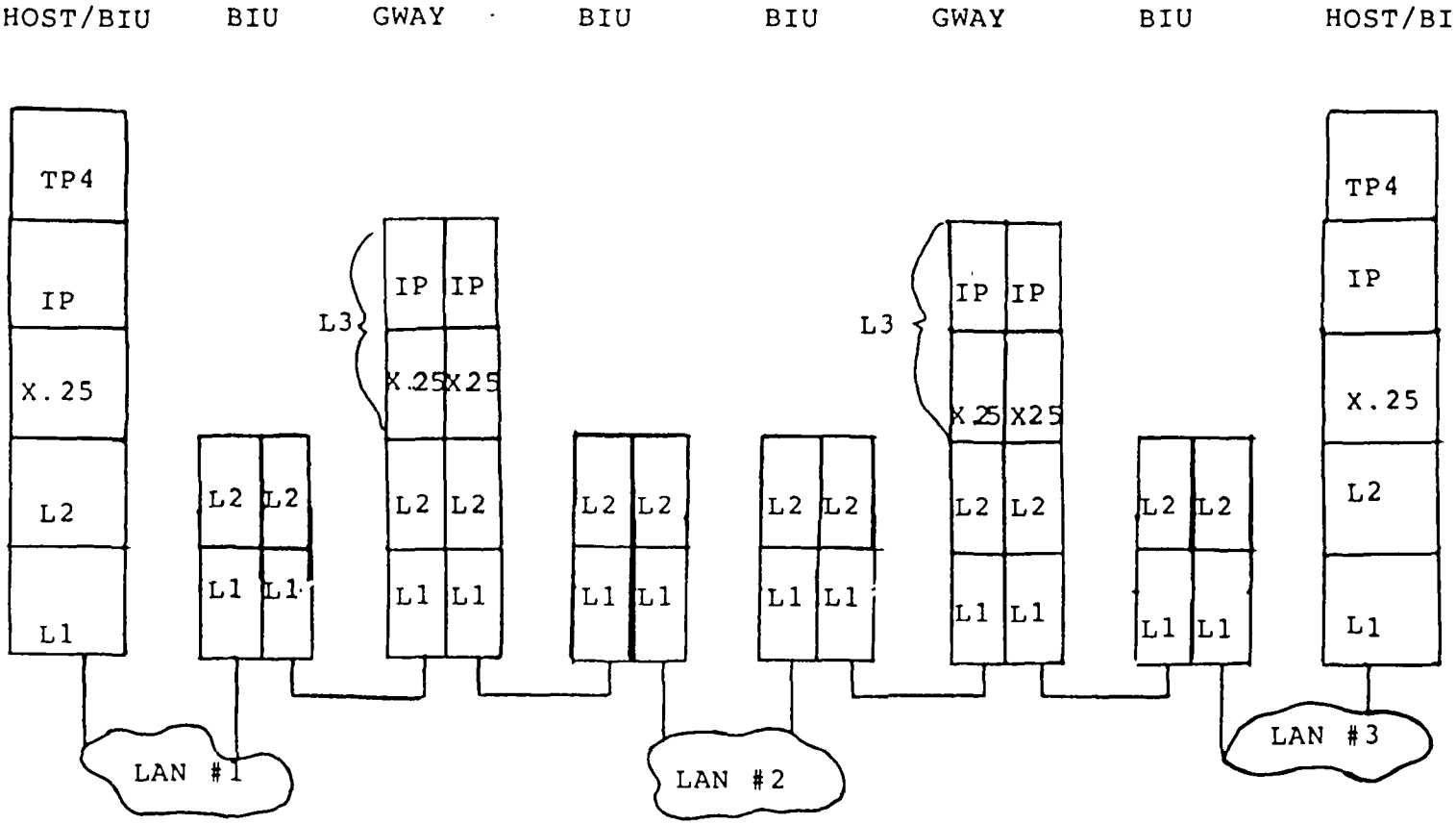


NOTES:

1. TP4= ISO Class 4 Transport Protocol
   IP = Connectionless Internetwork Protocol
   Li = OSI Protocol layer i
2. LAN's 1 and 2 are identical and employ connectionless oriented network layer protocol.
3. LAN 3 is incompatible. It too is connectionless at the network layer.
4. Gateway - BIU interface may be either serial or parallel

FIGURE 3-1a

OPERATIONAL PHASE PROTOCOL ARCHITECTURE



NOTES:

1. TP4= ISO Class 4 Transport Protocol
   IP = Connectionless Internetwork Protocol
   Li = OSI Protocol layer i

2. LAN's 1 and 2 are identical and employ connection oriented network layer protocol.

3. LAN 3 is incompatible. It may be connection oriented or connectionless at the network layer.

4. Gateway - BIU interface may be either serial or parallel.

5. Connection must be terminated when interfacing an incompatible network.

FIGURE 3-1b

Protocol layers 1 and 2 are unique to each LAN. The layer 1 and 2 protocols for the BIU to gateway connection, however, are assumed to be the same for all gateways connected to a given LAN. For example, the gateway may be interfaced to the BIU by means of either serial or parallel, synchronous or asynchronous I/O ports. Implementations may be as diverse as high speed direct memory access (DMA) channels or low speed RS-232 lines.

For the initial Demonstration phase, the layer 3 protocol for each LAN shall employ only a connnectionless oriented protocol. The concepts of the demonstration system described above are illustrated in Figure 3-2. In this example, a personal computer workstation (e.g., IBM PC) is connected to a FODS BIU via a serial I/O port. A more sophisticated host processor (e.g., Intellimac IN/7000) is connected to a JSC/FDDI LAN via a high speed interface. These two networks are coupled together by a full host gateway implementation which also uses HSI's to communicate with the FODS and JSC/FDDI LAN's. In this example, the Intellimac host wishes to transmit data to the PC workstation. The Intellimac builds packets of data which include the Internetwork Protocol header (see Section 4). To make maximum use of the JSC/FDDI bandwidth, the Intellimac generates packets 4500 bytes long (i.e., maximum JSC/FDDI packet size). The packets include an Internetwork Protocol header identifying the packet size and noting that they are not fragmented. Packets are routed to the JSC/FDDI BIU connected to the FODS-JSC gateway. The gateway accepts the packets and, by examining a LAN topology definition table, determines where to route the packets and that they must be fragmented to meet the smaller packet size

DATA FLOW JSC/FDDI TO FODS

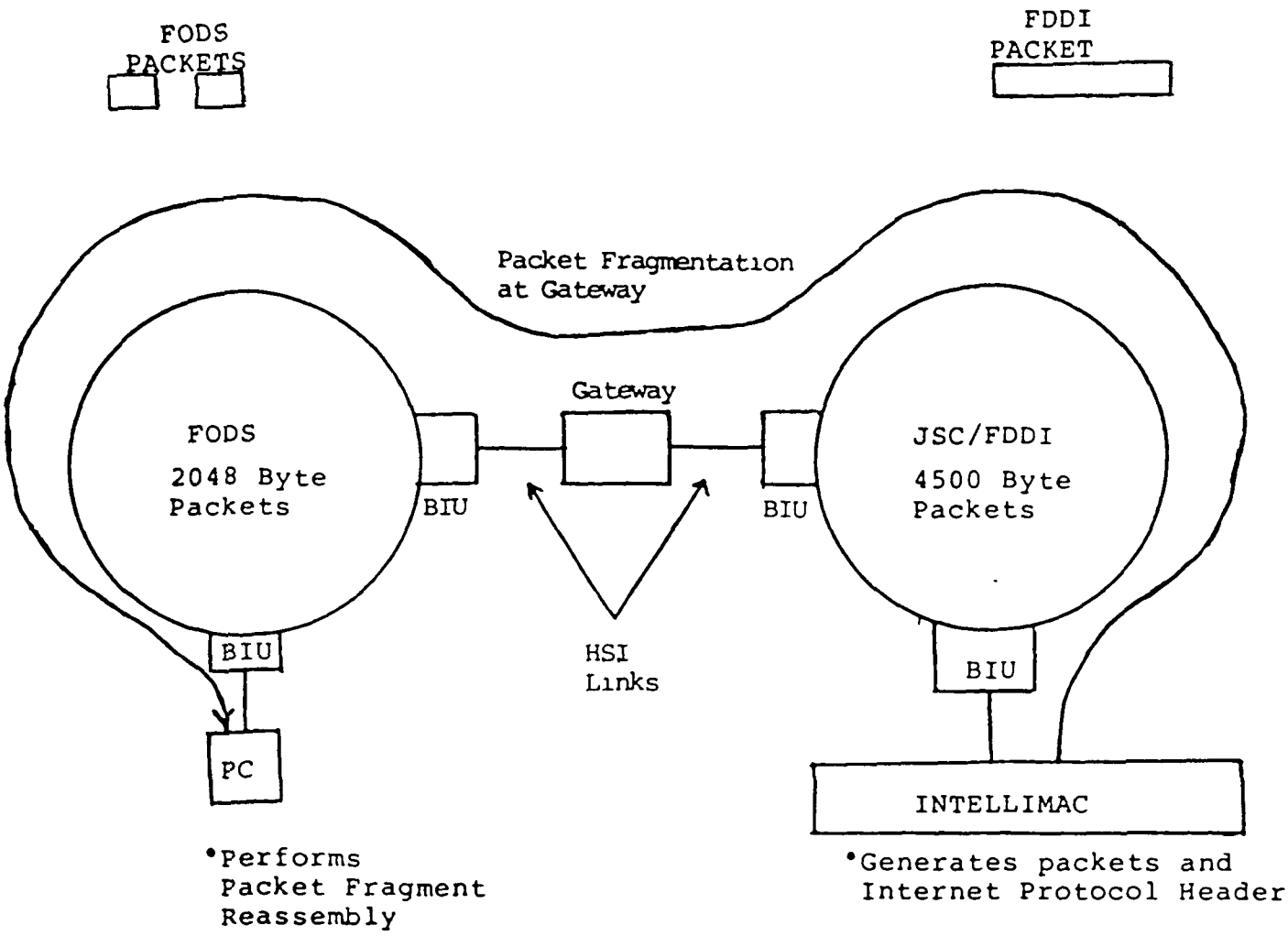


FIGURE 3-2

limitation of the FODS. The gateway, therefore, is responsible for fragmenting the 4500 byte packets into 2048 byte packets and generating fragment numbers and byte offset indices in the Internetwork Protocol headers of each packet. Once routed to and received by the PC workstation, the PC is then responsible for correctly reassembling the packet fragments based upon the Internetwork Protocol header information extracted from each packet fragment.

The layer 3 protocol for each LAN may differ, in the Operational phase. For example, in Figure 3-1, LAN's 1 and 2 are identical. Thus,they may employ a connection oriented protocol and utilize virtual circuits to create a path for packet transmission between nodes on the two LAN's. LAN 3, on the other hand, may use a datagram or connectionless protocol to route and forward packets. The mix of connection and connectionless protocols shall be accommodated by the gateway for the Operational phase. If two dissimilar LAN's which both employ virtual circuits are interconnected by the gateway, the individual virtual circuits shall be terminated in the gateway. Also, when a gateway interconnects a virtual circuit oriented LAN with a datagram oriented LAN, the virtual circuit service shall terminate at this juncture. Virtual circuit service shall be supported only when a contiguous path exists through LAN's and gateways which support this service.

Another condition affecting virtual circuit service is interconnecting two incompatible LAN's with different maximum packet sizes. In Figure 3-1, LAN's 2 and 3 are incompatible due to a difference in packet sizes accommodated by the respective

networks. The gateway interconnecting these incompatible networks must fragment each large packet from LAN 2 into multiple smaller packets to accommodate LAN 3. Due to this incompatibility between LAN's 2 and 3, the network layer must support a hybrid virtual circuit and datagram service. The latter shall be utilized to identify each fragment with a count or offset in order to properly sequence the data and assure proper packet reconstruction by the end destination host processor for whom they are intended. A new virtual circuit service must therefore be established at this point and be maintained until another incompatible network (e.g., connection oriented versus connectionless; or fragmentation required) is encountered.

Figure 3-3 illustrates a representative topology for multiply interconnected LAN's. The figure illustrates the following assumptions:

- gateways are implemented as independent processors (i.e., "full host" implementation versus interconnected BIU's forming two gateway halves) - see Appendix A.
- multiple LANs may be interconnected
- the gateway-LAN interface may differ among LAN's (i.e., one LAN may interface using a parallel mode while another may interface in a serial mode)
- there may be more than one gateway per LAN
- more than one path may be defined to enable one LAN to communicate with another (e.g., data from LAN 1 may be sent directly to LAN 3 or indirectly via LAN 2)
- a gateway may interconnect more than two LANs

LAN TOPOLOGY EXAMPLE

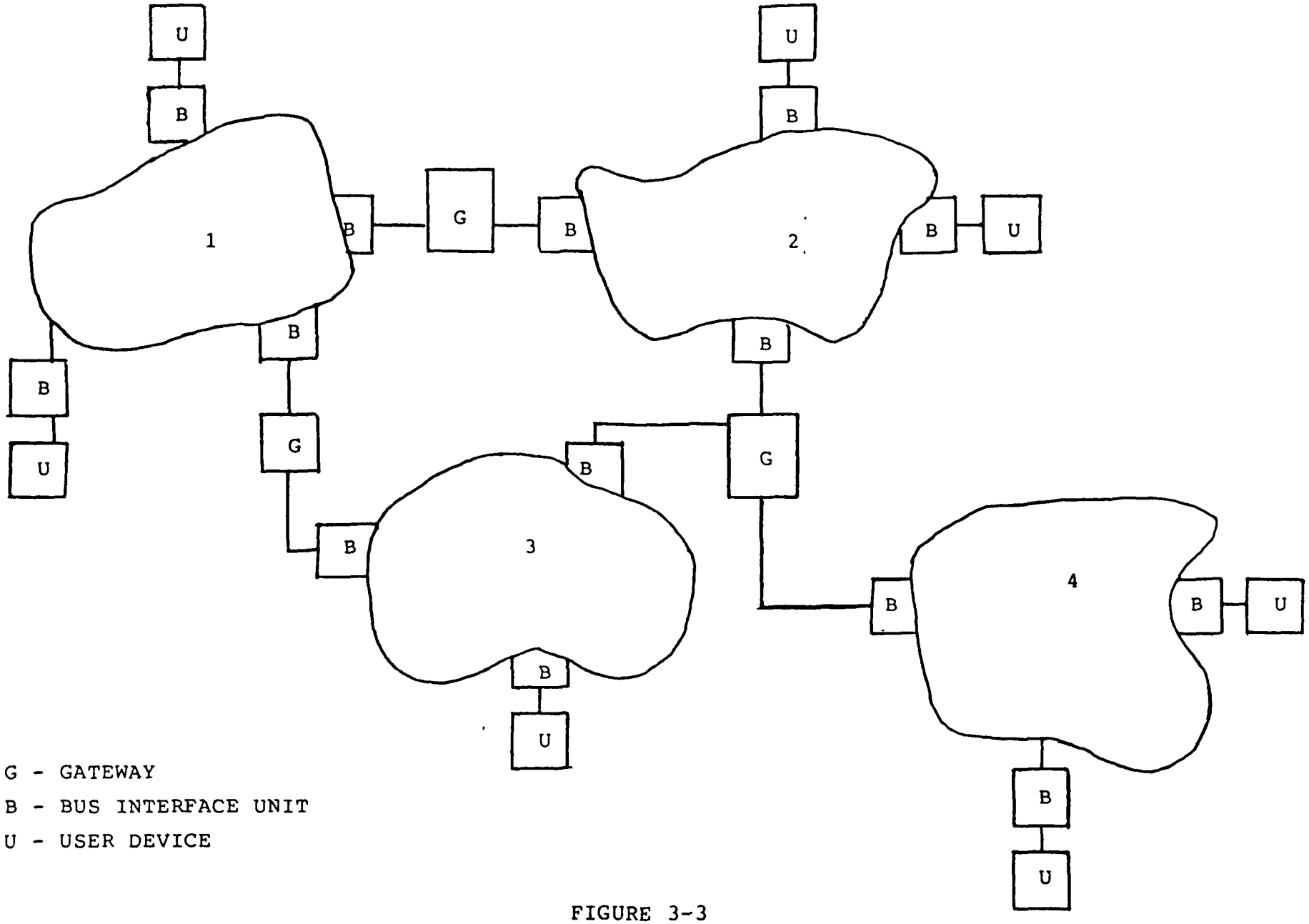


FIGURE 3-3

- transmission speeds of the LANs may differ

- LAN packet sizes may differ

- LAN protocol layers 1 and 2 may differ: the gateway shall reconcile these differences when exchanging packets between LANs

- layer 2 may provide highly reliable (i.e., low bit error rate) yet unchecked (i.e., no error detecting checksum) service between the BIU and gateway

BIU's shall be responsible for receiving data from and transmitting data to the fiber optic LAN. The BIU may be an integral component (e.g., printed circuit board) of the device it serves (e.g., host processor, intelligent workstation, or instrument), or an independent, stand-alone unit. In fact, its implementation may vary from one LAN node to another on the same LAN. Functionally, however, all BIU's within a particular LAN shall be identical. The interface between the BIU and the LAN is a network interface unit (NIU) which handles the fiber optics media access protocol. The interface between the BIU and the device it serves (e.g., gateway) may be either serial or parallel, synchronous or asynchronous, or support full or half duplex communication. However, it is assumed that only one such interface exists between a BIU and a gateway. A BIU may support several serial and/or parallel interfaces to other user devices (e.g., host computers). As a minimum requirement, it shall be .pa assumed that the BIU's are capable of interfacing with the Sperry

FODS, NASA/GSFC Code 700 LAN, and the NASA/JSC token ring LAN. Appendix B describes ther salient characteristics of the FODS High Speed Interface (HSI) and RS-232 interfaces to the user.

SECTION 4 - DEMONSTRATION INTERNETWORKING FUNCTIONALITY

Section 4 of this Design Specification focuses on the functional characteristics for a gateway Demonstration capability. The following gateway functions are addressed in the sections indicated below:

4.1 Routing and Data Transfer
4.2 Packet Fragmentation
4.3 Congestion Control
4.4 Network Management

Each section provides a brief narrative describing the function. This is then followed by an Input, Processing, Output specification supplemented with figures identifying relevant data structure contents.

4.1 Routing and Data Transfer

The gateway shall be responsible for performing packet routing using a connectionless (i.e., datagram) oriented network protocol. Any packet received at the gateway intended for a node on another LAN shall be processed and then forwarded via any viable alternative route.

INPUT:

- All packets
- LAN Topology Table (Figure 4.1-1)

PROCESSING:

The gateway shall receive, process, and forward all packets designated for a node on another LAN. The gateway shall examine the destination address in the Internetwork Protocol packet header, access the LAN topology table, and select a predesignated path on which to forward the packet. The gateway shall prepend this next local address to the packet, per the HSI specification in Appendix B, and forward it.

OUTPUT:

- Link address of the next gateway or
- packet to layer 2 protocol

Figure 4.1-1: LAN Topology Definition Table Contents

For each destination address:

- alternative gateway address(es) for next gateway "hop" to the end (i.e., far) destination
- characteristics of next LAN (i.e., maximum packet size)

4.2 Packet Fragmentation

The gateway shall be capable of fragmenting packets it receives in order to forward them to the next LAN which handles smaller sized packets.

INPUT:

- any data or control packet too large for forwarding to next LAN
- Internetwork Protocol Header from large packet
- LAN Topology Definition Table

PROCESSING:

When a gateway receives a packet, it shall examine its LAN Topology Definition Table (TDT) to determine if the received packet size can be accommodated by the next LAN. The TDT shall indicate whether the next LAN handles dissimilar packet sizes and thus require the gateway to invoke packet fragmentation processing. If packet fragmentation is required, it shall be the responsibility of the processor at the end destination node to correctly reassemble the original packet contents.

If the packet received by the gateway is equal to or smaller than the maximum packet size accommodated by the next LAN, then the gateway shall simply forward it as described in Section 4.1 - Routing and Data Transfer.

If the received packet is larger than the largest packet accommodated by the next LAN, the gateway shall implement a procedure to fragment the received packet into N outgoing packets. Each of the N outgoing packets (except possibly the last packet) shall be as large as the largest packet which can be accommodated by the next LAN less the amount of Internetwork

Protocol header overhead. The gateway shall record sufficient information in the Internetwork Protocol Header to enable the end destination node to correctly recombine all fragments of a packet. This information shall be comprehensive enough to account for the distinct possibility that packets may be received out of their proper sequence at the end destination node. The information contained in the Internetwork Protocol Header shall include the following, as a minimum requirement:

- the total length of the packet including both header and data fields
- a datagram number for the end destination node to determine which packet the fragment belongs to
- a field which identifies whether more fragments are to follow (i.e., this field is set to 'more fragments' for all fragmented packets except the last)
- a fragment offset which identifies the placement of the fragment within the original packet

Optionally, each fragment shall also include a checksum for the fragment header information field.

This is of special importance to assure the end destination node correctly reassembles all fragmented packets.

OUTPUT:

- N packet fragments with updated Internetwork Protocol Header

4.3 Congestion Control

The gateway shall invoke a congestion control algorithm to accommodate fluctuations in traffic loading and differences in LAN data rates. Congestion control shall be implemented by the gateway when it has been determined that too many packets saturate the gateway throughput capability. The strategy which shall be implemented is a gateway initiated message to flow control user devices.

INPUT:

- Buffer congestion index
- Table of active user processes

PROCESSING:

When the level of congestion in the gateway buffer has changed (i.e., significanly increased or decreased), the buffer manager shall pass to the gateway an updated buffer control index. This index shall take on a value of 0 to N (TBD) indicating the level of congestion in the gateway buffer.

The gateway shall prepare and forward an inter-LAN control message to a management process in each host via connectionless protocol service. This control message shall specify the buffer control index.

Upon receipt of the control message, the management process shall adjust flow control windows in their respective user devices to reflect the new buffer congestion index.

OUTPUT:

- Inter-LAN control message
- Updated flow control windows (in user devices)

4.4 Network Management

The gateway shall support network management functions. Network management in the gateway shall be utilized to (1) support systems management and (2) to perform layer management for the physical, data link, and network layers.

The demonstration gateway shall suppport one systems management function, statistics processing. Statistics shall be maintained by the gateway on buffer use, traffic volume and rates, and errors.

INPUT:

- system management REQUEST packet
- timer expiration from Timer Services
- timer request from statistics and data transfer
- buffer, traffic, and error information from data transfer

PROCESSING:

Upon timer expiration or a request from systems management, the gateway shall send statistics with counts on buffer use, number of packets, and number of errors for last T seconds (since previous timer expiration) or since the last systems management request.

OUTPUT:

- statistics information for systems management

SECTION 5 -OPERATIONAL INTERNETWORKING FUNCTIONALITY

Section 5 of this Design Specification focuses on the operational functional characteristics the gateway is required to support. The following gateway functions are addressed in the sections indicated:

- o 5.1 Routing
- o 5.2 Set-up Procedures
- o 5.3 Data Transfer
- o 5.4 Disconnect
- o 5.5 Packet Fragmentation
- o 5.6 Congestion Control
- o 5.7 Network Management
- o 5.8 Security

Each section provides a brief narrative describing the function. This is then followed by an Input, Process, Output specification augmented with tables, data flow diagrams, and other figures as necessary to fully describe each function.

5.1 Routing

The gateway shall be responsible for performing packet routing. All packet routing shall be over a virtual circuit established and maintained by the gateway until it receives and processes a CLEAR_REQUEST which disconnects a specific, active virtual circuit. The gateway shall have available all information required to identify the number and characteristics of the LAN's to which it is connected. The gateway shall implement a cyclic routing algorithm which selects the next valid alternative path for packet routing each time a virtual circuit is to be established. Once established, the gateway shall be responsible for determining the incoming virtual circuit number for any packet, and mapping that circuit into an outgoing virtual circuit number for packet forwarding.

INPUT:

- All packets
- Virtual Circuit Table (Figure 5.1-1)
- LAN Topology Table (Figure 5.1-2)
- Cyclic Routing Table (Figure 5.1-3)

PROCESSING:

The gateway shall receive and process CALL_REQUEST packets per the set-up procedures defined in Section 5.2. The selection of the virtual circuit path shall be determined by a cyclic routing algorithm implemented in the gateway. The gateway shall examine its LAN Topology Table and select the next viable path to the destination address specified in the CALL_REQUEST packet. For

example, if three viable paths exist to the specified destination address, then path 1 shall be used for the first CALL_REQUEST packet, and paths 2 and 3 for the subsequent two CALL_REQUEST packets received by the gateway specifying the same destination address.

OUTPUT:

- Updated Cyclic Routing Table
- Link address of the next gateway

Figure 5.1-1: Virtual Circuit Routing Table Contents

For each incoming virtual circuit:

- Host address (network ID, BIU address, port number)
- Host virtual circuit number

For each outgoing virtual circuit:

- Host address (network ID, BIU address, port number)
- Host virtual circuit number

Figure 5.1-2: LAN Topology Table Contents

For each destination address:

- alternative gateway address(es) for next gateway "hop" to the end destination address

Figure 5.1-3: Cyclic Routing Table

For each end destination address:

- list of alternative viable "hops" to next gateway
- index to next "hop" to be used by gateway

5.2 Set-up Procedures

A virtual circuit connection shall be established and maintained for node-to-node communication. As such, the gateway shall perform certain set-up procedures before transmitting data to or receiving data from any other node. A mechanism utilizing CALL_REQUEST and CALL_CONFIRMATION packets shall be used for this purpose. Successful completion of the handshake completes the set-up procedure and enables virtual circuit communication between the two nodes. These setup packets could be based on either X.25 or the ISO Class 4 Transport protocol.

INPUT:

- CALL_REQUEST packet (Figure 5.2-1)
- CALL_CONFIRMATION packet (Figure 5.2-2)
- REJECT packet (Figure 5.2-3)
- Virtual Circuit Table (Figure 5.2-4)
- Throughput Control Table (Figure 5.2-5)
- CLEAR_REQUEST packet (Figure 5.2-6)
- CLEAR_CONFIRMATION packet (Figure 5.2-7)
- packet expiration timer

PROCESSING:

When a virtual circuit is to be set up, the gateway shall receive, process and forward a CALL_REQUEST packet. Upon receipt of a CALL_REQUEST packet, the gateway shall attempt to allocate a virtual circuit using its Virtual Circuit Table (VCT). If a virtual circuit cannot be allocated, the gateway shall attempt to preempt one of the virtual circuits in use. Preemption shall be attempted for the virtual circuit which has a service priority

lower than that of the incoming CALL_REQUEST packet. If there are no candidates for preemption, the gateway shall issue a REJECT packet to the source address specified in the CALL_REQUEST packet and then exit this processing state. The reason for the reject (i.e., lack of virtual circuit resource) shall be identified in the REJECT packet.

If the capacity to support a virtual circuit is available, the gateway shall flag that it is temporarily allocated in the Virtual Circuit Table. Buffer space shall also be temporarily allocated at this time. The gateway shall also examine the service class field within the CALL_REQUEST packet. The gateway shall compare the service class requested to the total throughput currently allocated in the Throughput Control Table (TCT). If the addition of the requested service class does not exceed the maximum throughput for the gateway, the gateway shall add this request to the TCT. The gateway shall then forward the CALL_REQUEST packet to the next LAN. Subsequent receipt of the CALL_CONFIRMATION packet, in the return direction, by the gateway shall cause the virtual circuit and associated buffer space to be permanently allocated for the duration of the virtual circuit.

If the requested service class exceeds the total throughput available, the gateway shall examine the TCT entries to determine if an ongoing virtual circuit can be preempted and issue a CLEAR_REQUEST packet to tear-down one of the virtual circuits. The CLEAR_REQUEST packet shall be issued to one of the two nodes communicating over a virtual circuit selected by the gateway. A field within the CLEAR_REQUEST packet shall identify the reason

for this tear-down request. The gateway shall temporarily allocate the virtual circuit in the VCT, the required buffer space, and the requested service class in the TCT for the CALL_REQUEST packet. The CALL_REQUEST packet shall then be forwarded to the next LAN after the gateway has received a CLEAR_CONFIRMATION packet from the reverse direction.

The virtual circuit for which the CLEAR_REQUEST was issued shall be chosen by an algorithm which selects the virtual circuit with the least traffic volume and service class priority. It shall be the responsibility of the two end-point communicating nodes to complete the tear-down process for the virtual circuit.

Figure 5.2-8 illustrates the steps taken by the gateway for the various conditions described above.

OUTPUTS:

- CALL_REQUEST packet (forwarded)
- CALL_CONFIRMATION packet (forwarded)
- REJECT packet
- resource allocation (tentative)
- resource allocation (actual)
- updated Throughput Control table

Figure 5.2-1: CALL_REQUEST Packet Contents

Packet Type: CALL_REQUEST

Use: Virtual Call Set-up

Contents:

- control/data field packet identifier (i.e., control)
- control packet type field (i.e., CALL_REQUEST)
- virtual circuit number
- source address (network ID, node address, I/O port to user)
- destination address (network ID, node address, I/O port to user)
- service class request
- service priority
- service type (full/half duplex)
- maximum packet size to be transmitted

Figure 5.2-2: CALL_CONFIRMATION Packet Contents

Packet Type: CALL_CONFIRMATION

Use: To trigger permanent allocation of virtual circuit and associated buffers within the gateway

Contents:

- control/data field packet ID (i.e., 'control')
- control packet type (i.e., CALL_CONFIRMATION)
- virtual circuit number
- source address - network ID, node ID, I/O port to user
- destination address - network ID, node ID, I/O port to user
- service class request
- service priority
- service type (full/half duplex)

Figure 5.2-3: REJECT Packet Contents

Packet Type: REJECT

Use: Gateway cannot set up virtual circuit

Contents:

- control/data field packet ID (i.e., 'control')
- control packet type field (i.e., REJECT)
- virtual circuit number
- source address - network ID, node ID, I/O port to user
- destination address - network ID, node ID, I/O port to user
- service class request
- service priority
- service type (full/half duplex)
- reject code - i.e., reason for the reject such as:
    - virtual circuit unavailable
    - service class throughput requirements unavailable
    - CALL_CONFIRMATION receipt timeout

Figure 5.2-4: Virtual Circuit Table Contents

For each virtual circuit:

- Incoming BIU address (Network ID, Node, and Port)
- Corresponding input virtual circuit number
- Outgoing BIU address (network ID, NOde, and Port)
- Corresponding output virtual circuit number

Figure 5.2-5: Throughput Control Table Contents

For each I/O port supported by the gateway:

- Maximum Throughput Rating (bits or Bytes per second)
- Number of Virtual Circuits
- Estimated Throughput Requirement for each Virtual Circuit (i.e., service class)
- Data Priority for each Virtual Circuit
- Full/Half Duplex capability

Figure 5.2-6: CLEAR_REQUEST Packet Contents

Packet Type: CLEAR_REQUEST

Use: To initiate a tear-down of a virtual circuit currently in use

Contents:

- control/data field packet (i.e., 'control')
- control packet type (i.e., ' CLEAR_REQUEST')
- virtual circuit number
- source address - network ID, node ID, I/O port to user
- destination address - network ID. node ID, I/O port to user
- service class
- service priority
- service type (i.e., full/half duplex)

Figure 5.2-7: CLEAR_CONFIRMATION Packet Contents

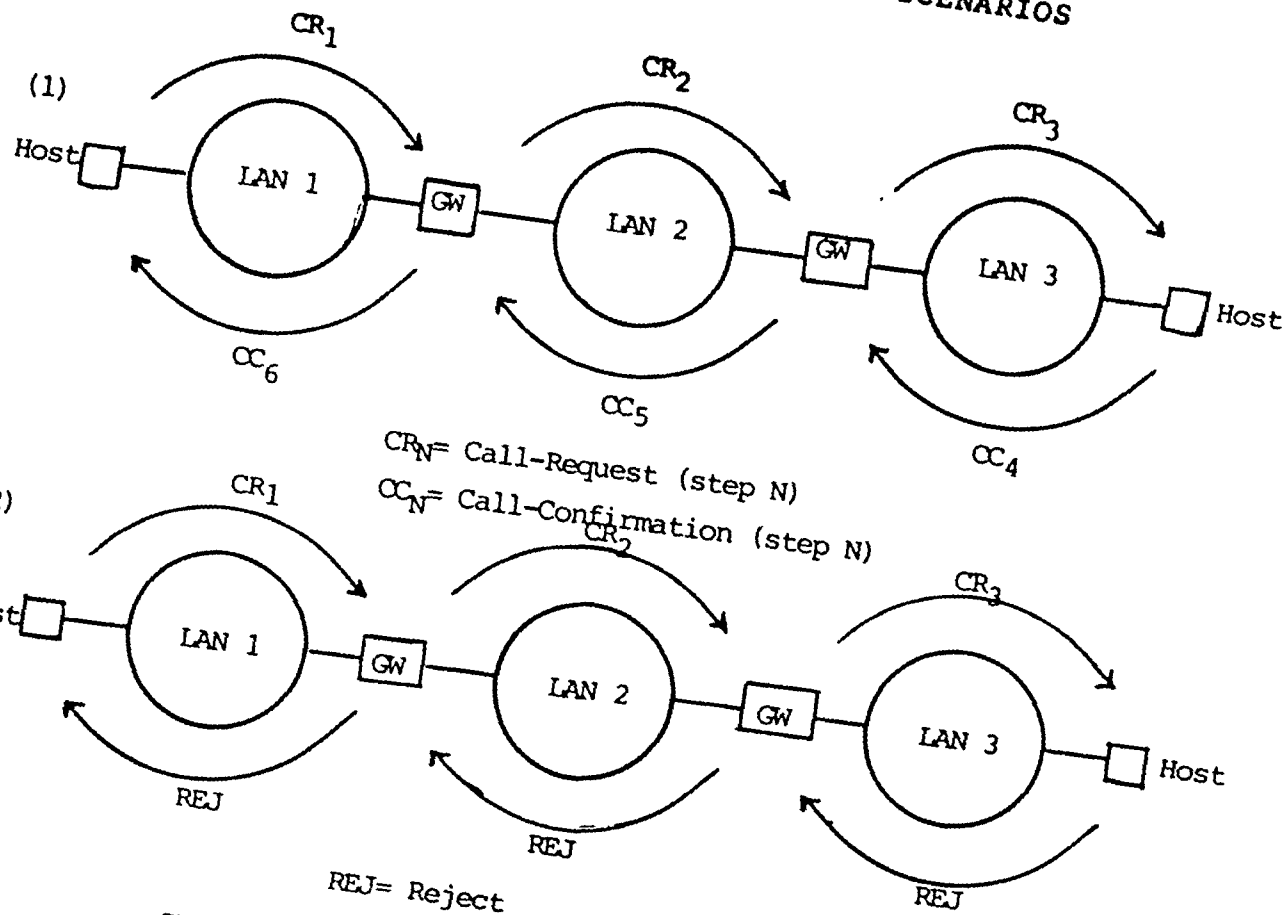
Packet Type: CLEAR_CONFIRM

Use: To confirm tear-down of virtual circuit when gateway has preempted its use for higher priority user.

Contents:

- control/data field packet (i.e., 'control')
- control packet type (i.e., 'CLEAR_CONFIRMATION')
- virtual circuit number
- source address - network ID, node ID, I/O port to user
- destination address - network ID, node ID, I/O port to user
- service class request
- service priority
- service type (i.e., full/half duplex)

GATEWAY CALL_REQUEST PROCESSING SCENARIOS



CR_N= Call-Request (step N)

CC_N= Call-Confirmation (step N)

REJ= Reject

Figure 5.2-8

5.3 Data Transfer

The gateway shall be capable of receiving and forwarding packets from one LAN to the another.

INPUT:

- DATA_PACKET
- control packets (e.g., acknowledgements)
- Virtual Circuit Table

PROCESSING:

The gateway shall receive and detect DATA_PACKETs and control packets (e.g., acknowledgements). Upon receipt, the gateway shall: examine the virtual circuit number of the incoming packet; determine the outgoing virtual circuit number from the VCT; change the virtual circuit address for the next LAN; prepend the local destination address for the next gateway or user node; and then forward the packet.

OUTPUT:

- DATA_PACKET
- control packets

5.4 Disconnect

The gateway shall be capable of receiving, recognizing, and processing CLEAR_REQUEST packets. These packets shall be used to tear-down the virtual circuit specified by the CLEAR_REQUEST packet.

INPUT:

- CLEAR_REQUEST packet
- Virtual Circuit Table
- Throughput Control Table

PROCESSING:

When the gateway receives a CLEAR_REQUEST packet, it shall initiate a process to tear-down the virtual circuit specified within the packet. The gateway shall temporarily deallocate the specified virtual circuit and associated buffers. It shall also flag the entry in the TCT to reflect its potential to honor higher throughput CALL_REQUEST packets. Upon receipt of a CLEAR_CONFIRMATION packet for the corresponding virtual circuit number or upon expiration of a timer for the CLEAR_CONFIRMATION packet receipt, the gateway shall proceed with the deallocation of buffers and the removal of the virtual circuit entry from the Virtual Circuit Table. The TCT shall similarly reflect the deletion of the virtual circuit and a higher gateway throughput availablity.

OUTPUT:

- temporarily/permanently deallocated virtual circuit
- temporarily/permanently deallocated buffers
- CLEAR_REQUEST packet forwarded to next LAN
- initiation of CLEAR_CONFIRMATION expiration timer

5.5 Packet Fragmentation

The gateway shall be capable of fragmenting packets it receives in order to forward them to the next LAN which handles smaller sized packets.

INPUT:

- any packet too large for forwarding to next LAN
- LAN Topology Definition Table

PROCESSING:

When the gateway receives a packet, it shall examine its LAN Topology Definition Table (TDT) to determine if the received packet size can be accommodated by the next LAN. The TDT will indicate whether the next LAN handles dissimilar packet sizes and thus require the gateway to invoke packet fragmentation processing.

If the packet received by the gateway is equal to or smaller than the maximum packet size accommodated by the next LAN, then the gateway simply forwards the packet without fragmenting it.

If the received packet is larger than the largest packet accommodated by the next LAN, the gateway shall implement a procedure to fragment the received packet into N outgoing packets. Each of the N outgoing packets shall be as large as the largest packet which can be accommodated by the next LAN less the amount of required header information overhead. The gateway shall prefix each fragmented packet with a header containing sufficient information to enable the far destination host processor to correctly recombine all fragments of a packet. This information shall include, as a minimum:

- the total length of the packet including both header and data fields
- a datagram number for the host to determine the fragment to which the packet belongs
- a field identifying whether more fragments are to follow (i.e., set for all fragments except for the last)
- a fragment offset to identify the placement of the fragment within the original packet

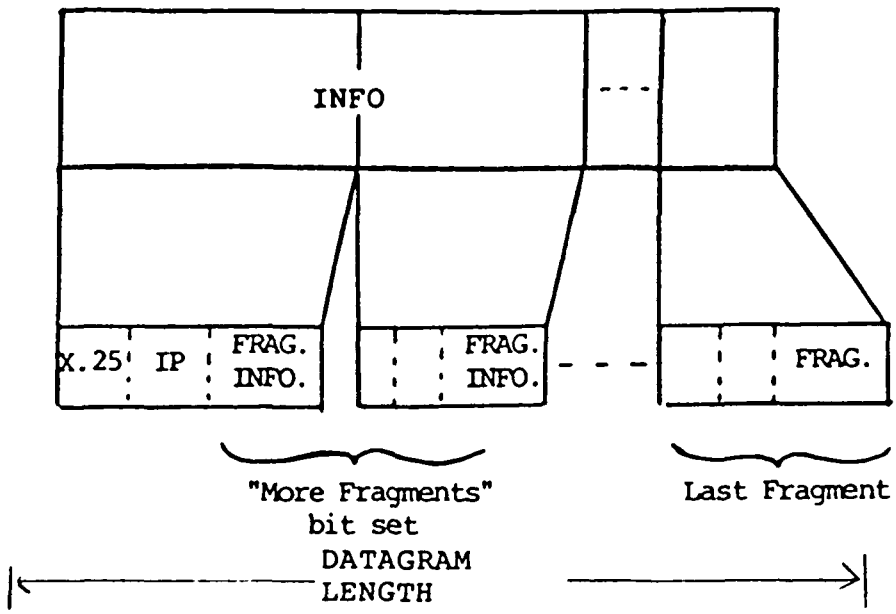
Optionally, each fragment shall also include a checksum for the fragment header information field.

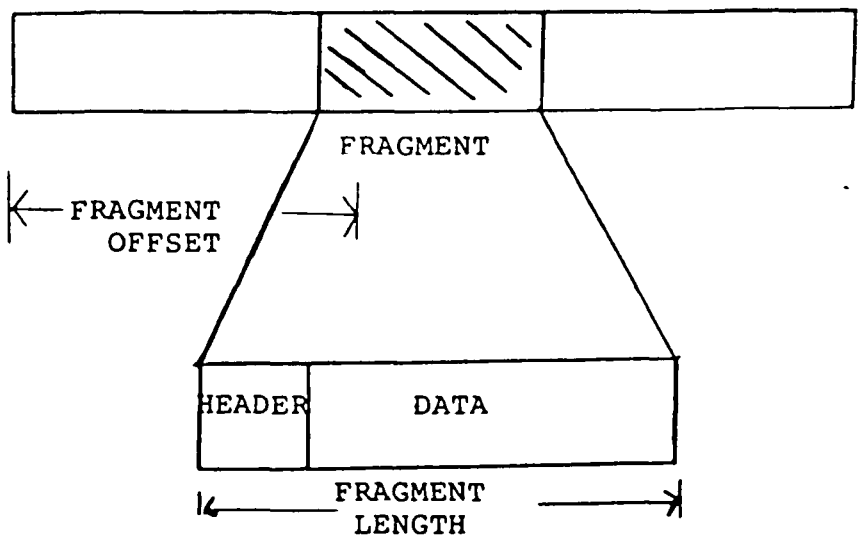
The process of packet fragmentation is illustrated in Figure 5.5-1.

OUTPUT:

- N packet fragments with packet reassembly information forwarded to the next LAN

Figure 5.5-1: Packet Fragmentation Processing Example





- DATAGRAM NUMBER
- DATAGRAM LENGTH
- FRAGMENT OFFSET
- FRAGMENT LENGTH
- SOURCE NETWORK ADDRESS
- DESTINATION NETWORK ADDRESS

5.6 Congestion Control

The gateway shall invoke a congestion control algorithm to accommodate fluctuations in traffic loading and differences in LAN data rates. Congestion control shall be implemented by the gateway when it has been determined that too many packets saturate the gateway throughput capability. The strategy which shall be implemented is a gateway initiated message to flow control user devices.

INPUT:

- Buffer congestion index
- Table of active user processes

PROCESSING:

When the level of congestion in the gateway buffer has changed (i.e., significanly increased or decreased), the buffer manager shall pass to the gateway an updated buffer control index. This index shall take on a value of 0 to N (TBD) indicating the level of congestion in the gateway buffer.

The gateway shall prepare and forward an inter-LAN control message to a management process in each host which has a user process with an active virtual circuit through the gateway. This control message shall specify the buffer control index.

Upon receipt of the control message, the management process shall adjust flow control windows in their respective user devices to reflect the new buffer congestion index.

OUTPUT:

- Inter-LAN control message
- Updated flow control windows (in user devices)

5.7 Network Management

The gateway shall support network management functions. Network management in the gateway shall be utilized to (1) support systems management and (2) to perform layer management for the physical, data link, and network layers.

The gateway shall suppport two forms of systems management: diagnostics and statistics processing. Diagnostics shall facilitate hardware fault detection and isolation as well as loop back tests to check for circuit continuity and integrity. Statistics shall be maintained by the gateway on buffer use, traffic volume and rates, and errors.

Layer management functions to be provided in the gateway shall include: resource allocation and timer services.

INPUT:

- system management REQUEST packet
- timer expiration from Timer Services
- timer request from statistics and data transfer
- buffer, traffic, and error information from data transfer

PROCESSING:

Upon timer expiration or a request from systems management, the gateway shall send statistics with counts on buffer use, number of packets, and number of errors for last T seconds (since previous timer expiration) or since the last systems management request.

Upon request from systems management, the gateway shall execute specified diagnostics and return results to systems

management. The gateway shall, upon request from systems management, loop back packets to systems management. Loop back shall be supported in either of two modes:

- one-shot packet loop-back (i.e., gateway loops-back the control packet requesting loop-back)
- continuous packet loop-back (i.e., the gateway loops back all packets from the source address until instructed to do otherwise)

Figure 5.7-1 illustrates the loop-back process. Thus, the gateway shall be capable of maintaining a virtual circuit with the systems management process.

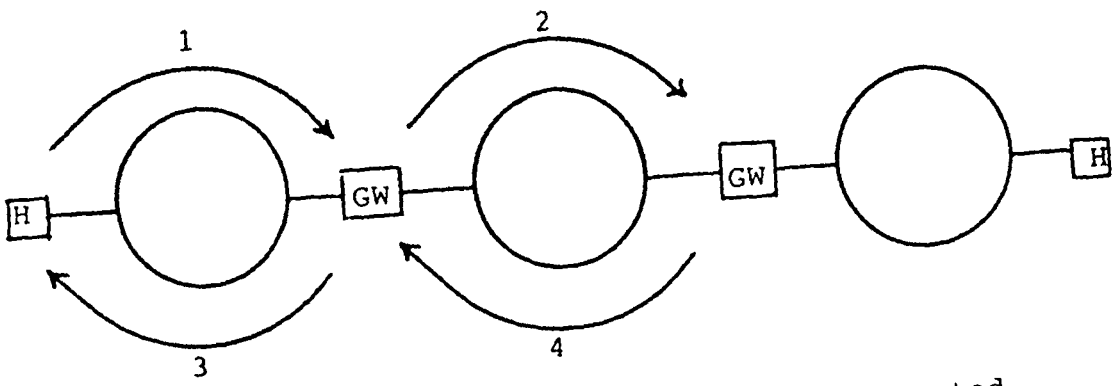
Upon request from set up, the gateway shall provide requested resources and ID or indicate that they are not available to the set up. The gateway shall also provide routing data by using the LAN toplogy table.

The gateway shall notify statistics and data transfer when a desired time interval has elapsed.

OUTPUT:

- diagnostic results to systems management
- loop-back packet(s) for systems management
- statistics information for systems management
- resource ID/ permission to set up (and indirectly to data transfer, fragmentation, and routing)

Figure 5.7-1: Loop-back Processing



Control Pocket with Loop-back Requested

5.8 Security

The gateway shall incorporate a minimal form of security. Security shall be incorporated in the gateway to assure that only authorized node pairs are permitted to communicate.

INPUT:

- CALL_REQUEST packet
- Gateway security access matrix (Figure 5.8-1)

PROCESSING:

The gateway shall honor a CALL_REQUEST set up packet only after it has found an entry in its GWSECURE matrix (Figure 5.8-1) granting CALL_REQUEST permission. If the gateway is unable to grant access between the two nodes, it shall return a "DENIED" condition code to the calling routine indicating a security access violation. The gateway set-up processing (Section 5.2) shall issue a REJECT packet to the source address and include the "DENIED" condition code as the reason for the rejection. Otherwise, the gateway shall return a "GRANTED" condition code and proceed with the CALL_REQUEST processing as defined in Section 5.2.

OUTPUT:

- Condition code indicating security access "GRANTED" or "DENIED"

Figure 5.8-1: GWSECURE Security Access Matrix

SOURCE NETWORK ADDRESS \ DESTINATION NETWORK ADDRESS	A	B	C	D	E	F
A		x				
B	x					
C	x	x		x	x	x
D	x	x	x		x	x
E	x	x	x	x		x
F	x	x	x	x	x	

NOTES: 1. LAN's C,D,E, and F may communicate with all other LAN's
2. LAN's A and B may only communicate with each other

APPENDIX A - GATEWAY ARCHITECTURE ALTERNATIVES

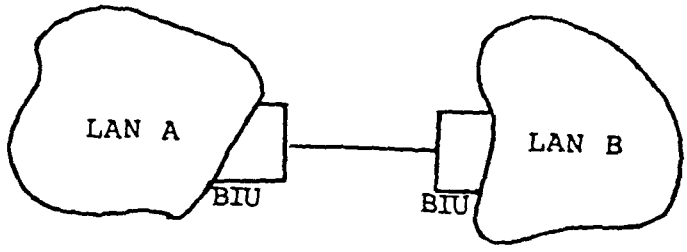
This appendix summarizes the information supplied in this appendix supplies the rationale for the selection of the gateway architecture specified in Sections 4 and 5 of this Design Specification. Various alternative architectural implementations for interconnecting heterogeneous LAN's are described.

Two candidate architectures for internetworking are the "full host" and the "half host" gateway. These architectures are illustrated in Figure A-1. In the full host implementation, the gateway is an independent host processor connected to Bus Interface Units (BIU's) on the LAN's it services. This design may be implemented such that the gateway is a single host computer interconnecting two or more LAN's. Alternatively two BIU's, one from each of two LAN's, directly coupled without any intervening host processor form a half host gateway implementation. In this case, the gateway is a shared resource between the two LAN's. Since an intervening host does not exist in this architecture, higher performance than a full host implementation will be realized. Additional host hardware is eliminated in this architecture which can result in a cost savings. On the other hand, each BIU half acting as a gateway must now be enhanced with software modifications to provide the gateway functions. Hardware modifications, such as additional memory for buffering, and possibly a higher performance processor may be likely. In summary, the advantages of the full host gateway over the half host gateway are:

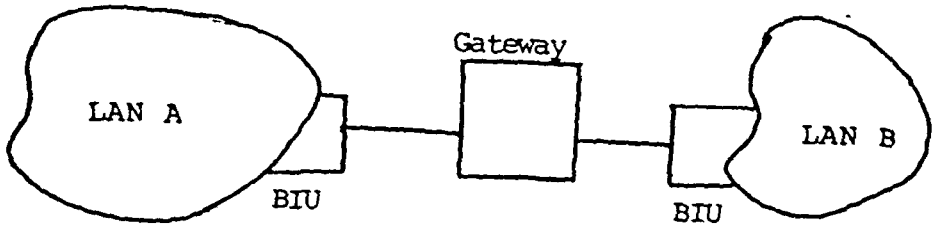
ALTERNATIVE GATEWAY ARCHITECTURES

FULL VS HALF HOST

FIGURE A-1



"Half-Host"



"Full-Host"

- the gateway now appears as a host processor to each LAN it services,
- no hardware or software modification to the existing BIU's is required to implement this architecture.
- greater flexibility to accommodate various types of LAN's
- modular architectural approach
- architecture is not constrained to any particular BIU,
- full host may interconnect more than 2 LAN's,
- may accommodate either serial or parallel interfaces to the BIU.

The principal disadvantages for the full host implementation are:

- the introduction of a full computer system and the cost of acquiring and maintaining it,
- lesser performance than half host since additional host processing is required in the full host,
- from an administrative point of view, the question of who owns and operates the gateway also becomes an issue.

A candidate for implementing the full host gateway is illustrated in Figure A-2. The full host gateway is comprised of a CPU (e.g., Motorola 68000 or Intel 286), RAM, DMA support, and two HSI channels. Interboard communication is over a Multibus, VME, or similar commercially available bus. Packets received via one HSI board are buffered in RAM under DMA control, processed by the CPU, and then retransmitted to the next LAN via the second HSI port. An RS-232 port to a keyboard CRT may be used to monitor gateway activity and permit a degree of operator control over gateway operation.

PULL HOST GATEWAY IMPLEMENTATION

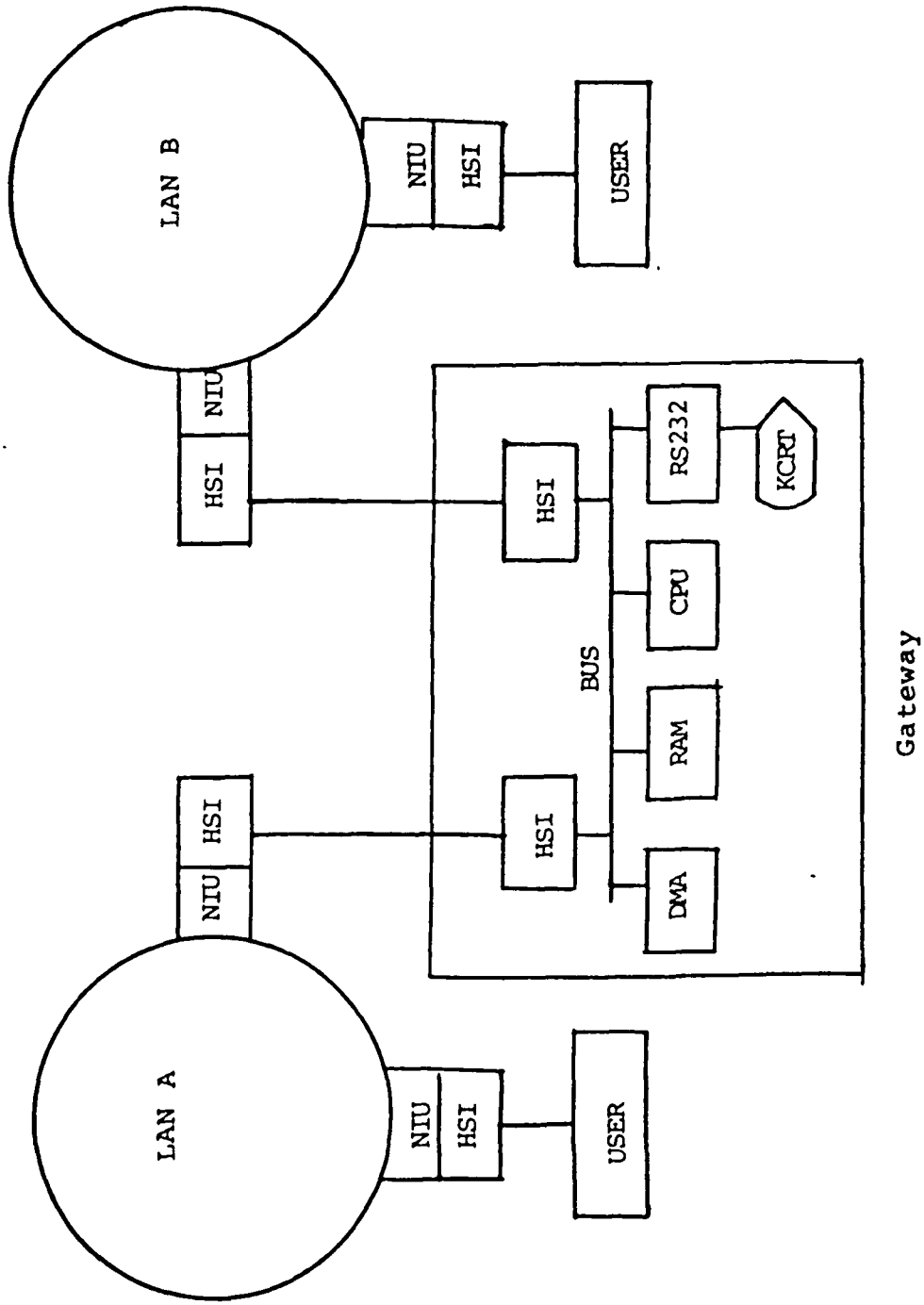


FIGURE A-2

It is important to note that a limitation exists in both gateway architecture alternatives in that a non-error controlled interface such as an HSI, is utilized to interface the BIU to the gateway. Thus, there is no error handling capability between pairs of HSI's. Data transmissions take place over a high speed, serial interface. Although the HSI bit error rate is low, an error in transmission is still possible. Data across this interface is thus unchecked. A gateway may receive a perfectly valid packet and in the course of forwarding it to the next LAN BIU, via its HSI port, introduce one or more bit errors. This limitation is not unique to the HSI, but to any unchecked I/O device. It is therefore an important consideration for any other interface implementation which does not append an FCS, checksum, or other data quality indicator when transmitting frames. To ameliorate this limitation, a software checksum could be incorporated into the network layer header. For example a cyclic redundancy checksum could be employed to detect such errors. However, this introduces significant processing load in the gateway.

Independent of the specific architectural implementation, there exists certain issues which must be addressed for the gateway design. These issues concern routing, virtual circuit set-up procedures, data transfer, virtual circuit disconnect, packet fragmentation to accommodate dissimilar LAN's, congestion control, network management, and security. Each of these issues is discussed below.

The gateway must be capable of routing packets from the designated source address to the designated destination address.

Obviously this is not an issue if only one path exists between source and destination. This is probably the case in the demonstration system but not the operational system. However, when more than one path may be taken, the gateway must be capable of determining, selecting, and forwarding packets over one of these alternative paths. The gateway does not want to always select the same path or bias its packet routing to any particular path since it may very well cause a congestion problem for one LAN and under-utilize alternative LAN's. The gateway could make use of periodic statistics messages identifying the degree of traffic for each LAN. The gateway could therefore make the best decision to select the least congested route available. The principal trade-off is implementation time and cost versus the need for an optimal route selection. For the Demonstration gateway architecture, it is recommended that a random routing algorithm be implemented if multiple routes are available. This algorithm selects the next viable gateway 'hop' randomly and therefore distributes the packet traffic over more than one LAN. This method is consistent with a connectionless protocol since the datagrams are not guaranteed to arrive at the end destination node in order. The Operational gateway architecture shall employ a connection oriented protocol where possible. For this architectural implementation, a cyclic routing algorithm is recommended. This algorithm selects each viable alternative route for each CALL_REQUEST set-up.

Virtual circuit service is preferred over datagram service for the Operational gateway architecture. The principal reason

for this is to accommodate high speed service classes such as digitized video data. As such, virtual circuit service requires that the gateway perform certain set-up procedures in order to establish a virtual circuit number, and allocate buffers and other resources as necessary before it is able to receive and forward data and control packets. A standard upon which the virtual circuit handling can be based is the is the X.25 network layer protocol. Set-up requires the recognition, processing, and forwarding of CALL_REQUEST and CALL_CONFIRMATION packets to establish the two-way virtual circuit handshaking between nodes on different LAN's. Alternatively the ISO Class 4 transport protocol could also be used.

Once a virtual circuit is established, the gateway must then support the data transfer phase. The gateway must be capable of recognizing an incoming data or control packet and its associated virtual circuit and forwarding it to the next LAN via the outgoing virtual circuit number maintained in the gateway's virtual circuit table.

Once a virtual circuit is no longer required, the gateway must be capable of disconnecting the virtual circuit. The gateway must therefore be capable of recognizing and processing CLEAR_REQUEST and CLEAR_CONFIRMATION packets from either pair of communicating nodes. The CLEAR_REQUEST packet initiates a process in the gateway to remove the virtual circuit routing information in the virtual circuit routing table and the deallocation of buffers and other resources.

When a gateway must interface LAN's handling dissimilar packet sizes, the gateway must be capable of fragmenting large

packets into multiple smaller packets. A key implementation issue is to supply a sufficient amount of information with each fragment to assure that they can be correctly reassembled by the destination host processor. One mechanism to accomplish this is to introduce an additional network protocol sublayer. A standard which is successfully used to address this problem is the connectionless Internet Protocol. This header identifies the length of the entire packet to be transmitted as fragments. It also provides the offset into the packet of the first byte of each packet fragment.

Congestion control is another issue the gateway architecture must consider when too many packets saturate the gateway processor. The simplest solution to this problem is to flow control the user devices sending data through the gateway. By issuing a packet to the user devices indicating the level of congestion at the gateway, the user devices can then reduce the data flow through the gateway until the congestion is reduced. The gateway can then issue another congestion control packet to user devices when it can accommodate additional packet processing.

Network management includes processing capabilities to determine network traffic statistics, LAN operational status (i.e., up or down), diagnostic processing, and other similar functions. This function can easily become a large, sophisticated process for any network. For the initial operational gateway design, it is recommended that the gateway support limited .pa diagnostic and statistics processing. These two functions will

provide insight into implementation bottlenecks and overall performance.

Security is another function which can easily become a large, sophisticated gateway capability. Password access control, node-to-node authorization access control, and gateway data encryption/decryption are just some of the forms of security control the gateway may exercise. For the Operational gateway implementation, and in keeping with a simple architectural approach, it is recommended that the gateway maintain a node-to-node access authorization matrix. This matrix relates source node addresses to destination node addresses for all LAN's. An entry in the matrix provides a simple "yes"/"no" authorization for the gateway to honor a CALL_REQUEST packet before setting up a virtual circuit.

APPENDIX B - FODS INTERFACE CHARACTERISTICS

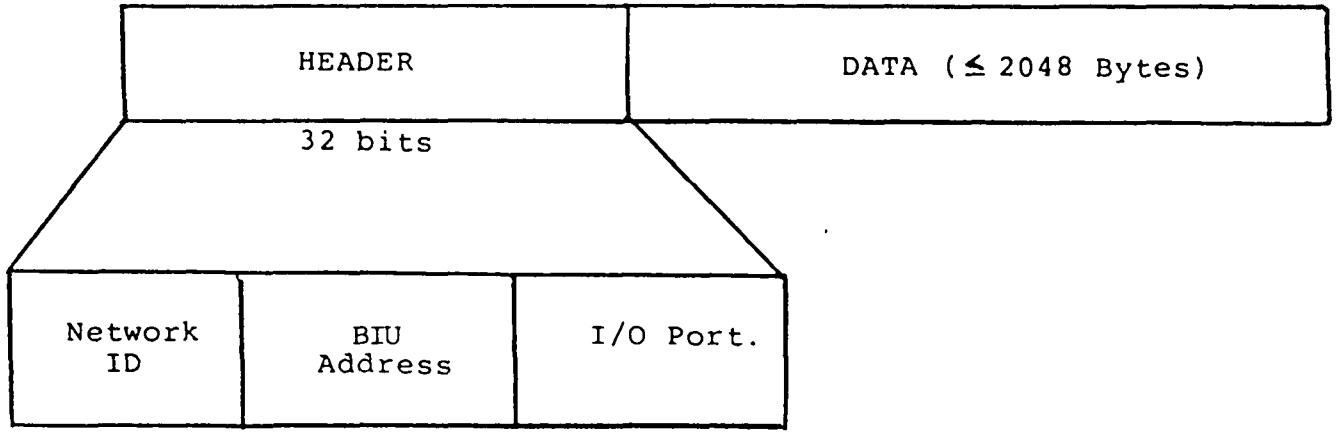
The interface between the FODS HSI and a user (e.g., host processor or gateway) consists of a header followed by up to 2048 byes of data. The header is 32 bits long and identifies the:

- network ID,
- BIU address, and
- port number

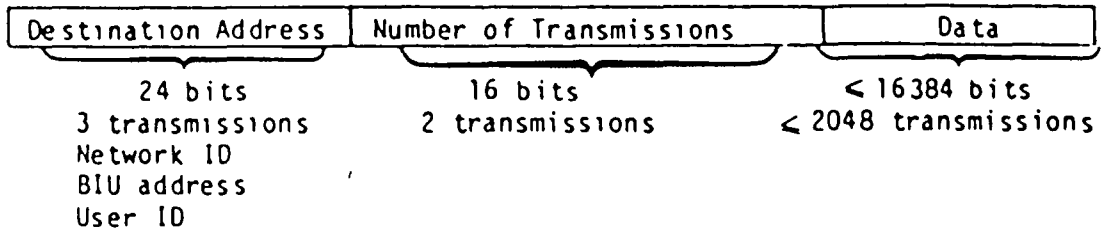
of the destination address for the packet. This address is the local destination address (i.e., "this" network) for the packet to be transmitted. Packets received by the HSI contain the same header information, but identify the local network source address for the received packet.

RS-232 packet formats contain the same header information as HSI packets. The header is immediately followed, however, by a count of the number of bytes in the subsequent information field.

These formats are illustrated in Figure B-1. Detailed descriptions may be found in the FODS System Specification referenced in Section 1 of this document.



a.) HSI Interface Format



b.) RS-232 Interface Format

Figure B-1: FODS Interface Format